Generative sampling with physics-informed kernels

Friederike Ihssen,[1] Renzo Kapust,[1, *] and Jan M. Pawlowski[1, 2]

[1] *Institute for Theoretical Physics, Universität Heidelberg, Philosophenweg 16, D-69120, Germany*
[2] *ExtreMe Matter Institute EMMI, GSI, Planckstr. 1, D-64291 Darmstadt, Germany*

We construct a generative network for Monte-Carlo sampling in lattice field theories and beyond, for which the learning of layerwise propagation is done and optimised independently on each layer. The architecture uses physics-informed renormalisation group flows that provide access to the layerwise propagation step from one layer to the next in terms of a simple first order partial differential equation for the respective renormalisation group kernel through a given layer. Thus, it transforms the generative task into that of solving once the set of independent and linear differential equations for the kernels of the transformation. As these equations are analytically known, the kernels can be refined iteratively. This allows us to structurally tackle out-of-domain problems generally encountered in generative models and opens the path to further optimisation. We illustrate the practical feasibility of the architecture within simulations in scalar field theories.

I. INTRODUCTION

Generative networks for Monte-Carlo simulations on high-dimensional data spaces with a given distribution face two out-of-domain (OOD) problems that hamper progress in this area.

The most obvious one is related to the high dimensionality of the data space that typically renders the learned distributions inaccurate. Roughly speaking this originates in the fact that interpolation problems in high dimensions are equally ill-conditioned as extrapolation problems in low dimensions. Consequently the computational costs of the final sampling steps with the true distribution scale badly with the dimension, which is a manifestation of the curse of dimensionality.

The second problem is a standard extrapolation problem and may be explained as follows: assume that the network has been trained up to a given sample size related to a limited information content: the size of a sample is in one-to-one correspondence to the access to a specific set of moments or cumulants of the distribution with the orders $n < N_{\max}$, where $N_{\max}$ increases with the sample size but scales very badly. For instance, for a scalar lattice field theory in two space-time dimensions, the dimensionality of the data space is given by $V = N_\tau \times N_s$, where N_τ is the number of lattice points in the (Euclidean) time direction and N_s is the number of lattice points in the spatial direction. In large scale lattice simulations, the lattice volume V easily reaches 10^6 to 10^8, depending on the space-time dimension.

Further sampling/training gives access to higher order moments/cumulants with $n > N_{\max}$. However, the information carried by the higher order moments, or rather their irreducible parts, is not contained in the lower ones. This information is provided by the final accept-reject step with the true distribution which hence is increasingly expensive: it scales as simply sampling with the true distribution in the first place. These two OOD problems lead to the bad scaling properties in the limit of large dimensions observed in many generative networks.

We hasten to add that the above problems are absent for distributions which have a finite number of independent moments after a reformulation within appropriate degrees of freedom but have a seemingly infinite number of independent moments in the given degrees of freedoms. If a generative network can unravel such a typically highly non-linear transformation within a rather finite data or training set, the generative task is practically solved.

In turn, the general situation asks for the construction of networks or sampling algorithms for which the problems above are solved constructively rather than hoping for the above underlying simplicity. Such a framework has to use the structural information that is encoded in the normalised statistical true distribution $p(\varphi)$,

$$p(\varphi) = \frac{1}{\mathcal{N}}\, e^{-\hat{S}(\varphi)} \to e^{-S(\varphi)}\,, \tag{1}$$

with the normalisation $\mathcal{N}$ and the exponential measure factor $\hat{S}(\varphi)$. In the second step in (1) we have absorbed the normalisation in a shift of the action, $S = \hat{S} + \log \mathcal{N}$. Then, S is nothing but the negative log likelihood and contains all the structural information of the distribution. In physics applications $\hat{S}(\varphi)$ is also called the action or effective Hamiltonian and φ is a field. For general applications, φ is a map from the set of data points $\mathcal{D}$ to the set of data values $\mathcal{T}$ and both sets may be discrete or continuous.

In the present work we suggest such a generative architecture based on physics-informed renormalisation group flows [1] (PIRGs) which aims at solving constructively the two OOD problems discussed above. PIRGs offer a comprehensive and general framework for invertible transformations of high-dimensional distributions with the RG-time t, analogously to the layerwise propagation and transformation in invertible networks. Importantly, all these steps follow tractable analytic differential equations which is key for the resolution of the OOD problems.

We proceed with a rough explanation of the construction principle with the example of a lattice field theory in mind. For this purpose we consider a deep network

[*] kapust@thphys.uni-heidelberg.de

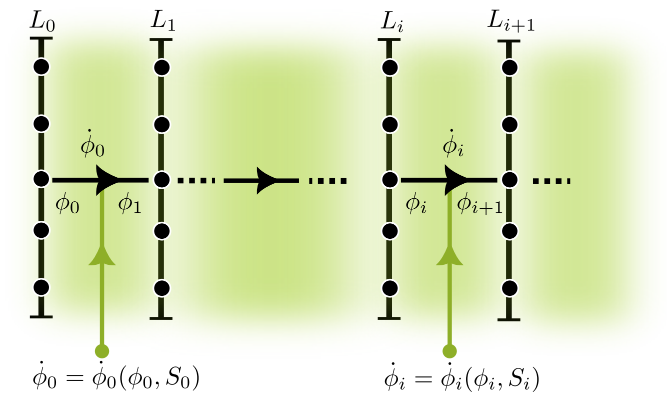

Figure 1. Illustration of the construction principle of PIKs for degrees of freedom ϕ that are propagated through the layers L_i using independent kernels $\dot\phi_i$.

of layers L_i with variables ϕ_i on the layer or rather the respective lattice. This is depicted in Figure 1.

The PIRG approach allows the construction of maps $\phi_i \to \phi_{i+1}$ on the basis of the pair $\{S_i(\phi_i), \phi_i\}$ of the negative log likelihood S_i and the field ϕ_i on a given layer L_i. The input layer L_0 has a chosen pair $\{S_0(\phi_0), \phi_0\}$. Natural choices for the input pair are the full action or negative log likelihood of the true distribution with $\phi_0 = \varphi$ and $S_0 = S$ or a Gaußian distribution with $S_0 = S_{\text{free}}$, where the subscript indicates a quadratic action. The kernels of these transformations are *physics-informed kernels* (PIKs). The respective transformation of the field ϕ_i on the layer L_i to the field ϕ_{i+1} on the layer L_{i+1} is given by a renormalisation group equation for the pair $\{S_i(\phi_i), \phi_i\}$, which is a diffusion equation for S_i: for the negative log likelihood S_i this is the Wegner equation [2], for other generating functions similar analytic RG equations exist, e.g. the generalised RG equation [3] for the rate function.

The key point of the PIRG approach is its use of the diffusive RG equation as a differential equation for the pair $\{S_i(\phi_i), \phi_i\}$ instead of a diffusive equation for S_i. For the present generative architecture this freedom is used as follows: Rather than following the diffusive dynamics and evolving the action S_i, all S_i's are fixed analytically. Then, the underlying renormalisation group equation is a simple linear differential equation for the kernel. This gives us an analytic access to the map between different layers in a deep network. Importantly, this linear differential equation does not contain information from the previous layers, thus completely decoupling the layerwise propagation steps from each other. With this physics-informed decoupling structure and determination of the layerwise map by an RG equation, the sampling network is to our knowledge qualitatively different from any other networks. The possible trivial parallelisation as well as the nature of the analytically known diffusion equation on the layers structurally resolve both of the OOD problems mentioned in the beginning. Furthermore, the underlying RG structure also allows for a systematic and optimisable upgrade of this procedure by also allowing for a learning of the set of actions $\{S_i\}$ and hence the construction of optimised PIKs, also accommodating improved or perfect actions in the RG procedure [4].

In the present work we briefly review PIRGs in the light of the present application to generative architectures in Section II. The general PIK architecture is put forward in Section III, where it is argued that PIKs allow for the construction of 'truly' generative networks. Here 'truly' refers to the fact that the probability of the final accept-reject step stays close to unity. The conceptual developments are showcased within a zero-dimensional lattice field theory as well as within first steps in higher dimensions in Section IV. We conclude in Section V with a summary and an outlook of some of the natural applications of PIKs. Some technical details are deferred to the Appendices, where also some of the machine-learning driven optimisation aspects are discussed.

II. PHYSICS-INFORMED RENORMALISATION GROUP FLOWS IN A NUTSHELL

Renormalisation group flows describe the scale-dependence and general reparametrisations of a given statistical distribution. While developed in quantum and statistical field theories, the concept is far more general. Both aspects are governed by the infinitesimal change of the distribution with the renormalisation time t. Accordingly, instead of the field φ in (1), we consider fields ϕ_t that are related to φ with a general non-linear transformation. Importantly, we do not have to resolve this transformation, but only its infinitesimal change from $\phi_t \to \phi_{t\pm\epsilon}(\phi_t)$. This transformation comes with a change of the distribution $p_t(\phi)$ and the respective differential equation is either derived for the measure $p_t(\phi)$ or its Laplace or Fourier transform

$$Z(J) = \int \mathcal{D}\phi\, p_t(\phi)\, e^{\sum\limits_{\hat n} \phi_{\hat n} J_{\hat n}}\,, \qquad \mathcal{D}\phi = \prod_{\hat n \in \mathcal{D}} \int_{\mathcal{T}_t} d\phi\,, \tag{2a}$$

where

$$p_t(\phi) = e^{-S_t(\phi)}\,. \tag{2b}$$

In (2) we have introduced the negative log likelihood $S_t(\phi)$, and the fields ϕ take values in $\mathcal{T}_t$. In general, $\hat n \in \mathcal{D}$ labels data points in the discrete or continuous data space. We have restricted ourselves to the case of discrete data points in (2), such as also present in our lattice field theory examples. There, data points are sites on a rectangular lattice with $\hat n = (n_1, n_2, ..., n_d)$ and $n_i \in [0, 1, ..., N_{i,\max}]$. The field $\phi_{\hat n}$ takes real values on each of these lattice sites. In the present work we use a (hyper-) cubic lattice with $N_{i,\max} = N_{\max}$ for all $i = 1, ..., d$. Then, the field is a map

$$\phi : \mathcal{D} \to \mathbb{R}^{\mathcal{D}}\,, \qquad \mathcal{D} = [0, 1, ..., N_{\max}]^d\,, \tag{3}$$

with $\phi \in \mathcal{C}$, and $\mathcal{C}$ is the space of maps (3) with a given norm, commonly the L_2-norm. Typical large lattice sizes range between $N_{\max} \propto 10^3$ in d=1,2,3 and $N_{\max} \propto 10^2$ for $d = 4$. The most general scale and reparametrisation RG transformation is accommodated by the flow of the measure

$$\frac{d\,p_t(\phi)}{dt} = \frac{\partial}{\partial\phi}\Big[\Psi_t(\phi)\,p_t(\phi)\Big]\,, \quad p_t(\phi) = e^{-S_t(\phi)}\,, \tag{4}$$

and was put forward in [2]. Recently, the Wegner flow has received attention within applications in generative models and its relation to optimal transport [5–9]. The flow in (4) is a total derivative and hence leaves the total distribution $Z(0) = 1$ invariant. In the absence of boundary terms and for $d\mathcal{T}_t/dt = 0$ this invariance is given by

$$\frac{d\,Z(0)}{dt} \equiv 0\,. \tag{5}$$

The case $d\mathcal{T}_t/dt \neq 0$ has been considered in [1] in the rate function formulation of PIRGs and is readily mapped to the case $d\mathcal{T}_t/dt = 0$.

An important subgroup of transformations are the choices $\Psi_t(\phi) \propto \partial S_t(\phi)/\partial\phi$ with the RG-time dependent classical action S_t. Then, (4) takes the form of a Fokker-Planck equation in high-dimensions, and is related to the Langevin evolution of the field (Stochastic Quantisation) and the Polchinski equation [10] for the Wilsonian effective action. Equation (4) with or without the source term carries redundancies which may slow down and complicate its numerical solution. For this and further structural, stability, and numerical convenience reasons one typically considers the rate function or effective action $\Gamma(\phi)$ obtained from a Legendre transform of $\ln Z(J)$. The respective generalised RG equation reads [3],

$$\left(\frac{d}{dt} + \dot\phi_t \frac{\partial}{\partial\phi}\right)\Gamma_t = \frac{1}{2}\,\mathrm{Tr}\left[\frac{1}{\frac{\partial^2\Gamma_t}{\partial\phi^2} + R_t}\left(\frac{d}{dt} + 2\frac{\partial\dot\phi_t}{\partial\phi}\right)R_t\right], \tag{6}$$

with the ϕ-independent scaling-kernel R_t and the field transformation $\dot\phi_t(\phi)$. For $\dot\phi_t = 0$ it reduces to the Wetterich equation [11], see also [12, 13]. If we set the change of the scaling kernel to zero, $dR_t/dt = 0$, the two equations (4) and (6) are related by the choice $\Psi \propto \dot\phi_t$. For more details see [1, 14, 15]. Note also that in the functional RG literature the total RG-time derivative d/dt is commonly written as ∂_t.

The key idea underlying physics-informed RG flows [1] is to exploit the full generality of the Wegner flow (4) for the distribution, the generalised flow (6) for the rate function, or related ones for other generating functions: Instead of solving the flows for a given $p_t(\phi)$, $\Gamma_t(\phi)$ or other generating function F_t, we keep the coordinates ϕ general and solve the equations for pairs

$$\big(F_t(\phi)\,,\,\dot\phi_t(\phi)\big)\,, \qquad \text{with} \qquad F_t \in \{S_t, \Gamma_t, ...\}\,. \tag{7}$$

Here, $\dot\phi_t(\phi)$ is the local change at a given RG-time t, and in the examples considered F_t is either the negative log likelihood or the rate function. For the present purpose we take the RG-time in a normalised time interval,

$$t \in I \qquad \text{with} \qquad I = [0, 1]\,. \tag{8}$$

Here, $t = 0$ is the initial time and $t = 1$ is the final time. In particular, (7) with (4) and (6) also accommodates solving (4) and (6) for $\dot\phi_t$ for a given negative log likelihood or action S_t or rate function Γ_t.

An illuminating example for a flowing pair $\big(p_t(\phi)\,,\dot\phi_t(\phi)\big)$ is the pair that leads to normalising flows in lattice field theories [16–19]. More specifically, we consider an inverse normalising flow with the Gaußian action $S_0 = S_{\text{free}}$ at $t = 0$ and the full action $S_1 = S$ at $t = 1$. This constitutes a map from a free Gaußian theory with the normalised distribution $p_{\text{free}}(\phi) = \exp\{-S_{\text{free}}(\phi)\}$ to the theory with full distribution $p(\varphi)$ in (1). In terms of cumulants, normalising flows map theories with infinitely many independent cumulants to a theory with only one independent cumulant, the second order one. Evidently, in this case the map accommodates an infinite amount of information. This is directly visible for the rate function Γ_t, where normalising flows constitute a flow from the rate function of the full theory with infinitely many non-trivial cumulants to a Gaußian, quadratic rate function with only one cumulant; all higher orders are absorbed in the map $\dot\phi$ via (6).

The PIRG setup with its analytic flows allows us to include the additional - infinite - amount of information within the sampling procedure beyond the training. The key novel feature of the architecture is a structural property of the physics-informed RG flows: If the *target action* $F_t \in \{S_t, \Gamma_t, ...\}$ is fully specified and known, the associated map $t \to F_t$ is completely independent of the previous ones and only depends on the target action at time t,

$$\dot\phi_t(\phi) = \dot\phi_t(\phi; F_t)\,. \tag{9}$$

Put differently, the *kernel* $\dot\phi_t$ of the map is *physics-informed*. This perspective has also been used in [8], where (9) is solved within an architecture with a global (for all t) loss function. In the present work we use that (9) leaves us with independent tasks of learning or solving $\dot\phi_t$ for the whole time line from t=0 to t=1. These separate tasks represent standard RG-steps and hence carry the respective physics information. They can be trivially parallelised and also allow for an independent iterative optimisation. Importantly, the setup structurally avoids the curse of dimensionality due to the analytic access.

We close this brief review with the remark, that in the present work we introduce and illustrate this novel simulation architecture using flows of $p_t(\phi)$ with the Wegner equation (4). The also highly relevant discussion of the qualitative reduction of redundancies achieved by using the rate function or related generating functions as well as the further key question of full optimisation within these models is deferred to future works.

III. SAMPLING ARCHITECTURE WITH PHYSICS-INFORMED KERNELS

In the present Section we use the PIRG setup to construct general flows with *physics-informed kernels* (PIKs) $\dot{\phi}_t$, (9), in the space of distributions. The general architecture of PIKs is put forward in Section III A at the example of the PIRG pair $(S_t, \dot{\phi}_t)$ and the Wegner flow (4). Possible parametrisations of PIRG pairs and solution strategies are discussed in Sections III B 1 and III B 2. Finally, we outline optimisation procedures in Section III C.

A. General architecture

With PIKs we aim at building a generative architecture that avoids the two OOD problems discussed in the introduction, which would enable the efficient sampling with sample sizes orders of magnitude beyond the sample size generated and used in the training. The architecture uses the PIRG pairs to enforce a path for the negative log likelihood or action S_t, leaving us with the task of computing or learning the respective map $\dot{\phi}_t$ from its analytically known differential equation.

In the following, we start from a simple base distribution $p(\varphi)$ at $t = 0$ and flow into a more complex target distribution at $t = 1$. Since the architecture is based on the flows (4) and (6), it is invertible as the flows are invertible. For the chosen flow direction, the partition sum at $t = 0$ reads

$$Z_0 = \int \mathcal{D}\varphi \, p(\varphi) \,, \qquad \phi_0 = \varphi \,, \tag{10}$$

and the flow of the field φ is given by

$$\frac{d\phi_t}{dt} = \dot{\phi}_t(\phi_t) \,, \qquad \phi_{t=0} = \varphi \,. \tag{11}$$

In (11), the physics-informed kernel vector field

$$\dot{\phi}_t : I \times \mathcal{C} \to \mathcal{C} \,, \tag{12}$$

characterises the transformation at each time $t \in I$ with the normalised RG-time interval $I = [0\,, 1]$. As noted before, the partition sum is invariant under the transformation (4) and hence Z_t is t-independent, see (5). We derive with (4) that

$$Z = \int \mathcal{D}\phi \, p_t(\phi) \,, \qquad \frac{dZ}{dt} = \int \mathcal{D}\phi \, \frac{d\, p_t(\phi)}{dt} = 0 \,. \tag{13}$$

In (13), the total derivative nature of (4) has been used as well as the assumption of the absence of boundary terms. The total change of field and action or negative log likelihood is covered by

$$\frac{dS_t(\phi)}{dt} + \dot{\phi}_t(\phi) \, \frac{\partial}{\partial \phi} S_t(\phi) = \frac{\partial}{\partial \phi} \, \dot{\phi}_t(\phi) \,, \tag{14}$$

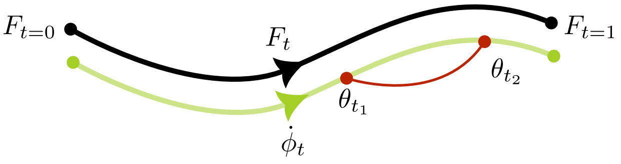

Figure 2. Training of most generative models from a distribution characterised by $F_{t=0}$ to one characterised by $F_{t=1}$. The path F_t is not fully determined and accessible. Optimisable parameters θ_t at different times are entangled, which is indicates by the red entanglement line between the parameters θ_{t_1} and θ_{t_2}.

which is the Wegner equation for the action, readily deduced from (4). Note that the right-hand side is nothing but the total derivative of $S_t(\phi_t)$. Here, $\partial/\partial\phi$ denotes the partial derivative w.r.t. each component in ϕ.

A key observation behind PIKs is the following: The PIRG setup (7) and (9) allows us to read (14) independently for each t as a simple linear first order differential equation for $\dot{\phi}_t(\phi)$ with a fully determined and accessible trajectory S_t. The kernel $\dot{\phi}_t$ then accommodates the change of the coordinate system in the data space that is induced by this trajectory. This structure also underlies other formulations of PIKs with $S_t \to F_t$, i.e. based on the PIRG pair with the rate function, $(\Gamma_t, \dot{\phi}_t)$. Then, (14) for $\dot{\phi}_t$ is substituted by (6) and interpreted as a simple first order differential equation for $\dot{\phi}_t(\phi; \Gamma_t)$, see the discussion around (9).

Accordingly, to our knowledge, the PIK architecture differs qualitatively from standard generative networks that aim to improve Monte Carlo simulations. This is due to the existence of accessible analytic maps for the layerwise propagation of the PIRG pair

$$(F_{t_i} \,, \dot{\phi}_{t_i}) \to (F_{t_{i+1}} \,, \dot{\phi}_{t_{i+1}}) \,, \tag{15}$$

with its independent solutions $\dot{\phi}_i(\phi; F_{t_i})$ with (6) and (14) or respective equations for general functions F_t.

In the following, we contrast the learning structure of standard generative approaches, depicted in Figure 2 with that of PIKs, depicted in Figure 3. Here the black line symbolises the path $S_t(\phi)$ connecting the base and targeted action. The green line illustrates the direction in which the kernel $\dot{\phi}_t$ with parameters θ_t is learned or computed.

For most generative models as in Figure 2 the path $S_t(\phi)$ is not known in a truly closed, analytically accessible form but is left open or is regressed from data using score-, flow- or Kullback–Leibler divergence oriented objectives [17–26]. The parameters θ_t of the kernel at different times t then usually depend on each other, and a change at one time affects the kernel at other times.

For PIKs in Figure 3 the path $S_t(\phi)$ is given in a fully determined and accessible form. Then, the PIK setup allows one to read the Wegner or continuity equation (14) as a simple linear first order differential equation for the kernel $\dot{\phi}_t$. This effectively splits up the global (and

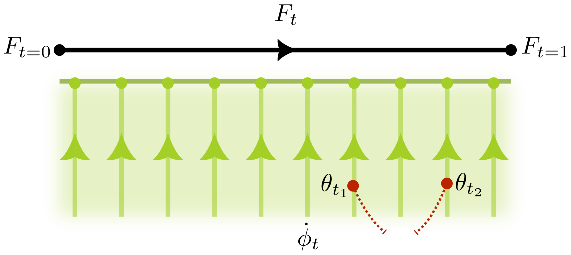

Figure 3. Training of PIKs from a distribution characterised by $F_{t=0}$ to one characterised by $F_{t=1}$. The path F_t is fully determined and accessible. Optimisable parameters θ_t at different times are not entangled, which is indicated by the broken dashed red lines.

possibly underdetermined) task from Figure 2 of finding a generative map between two distributions into a set of small fully determined and independent tasks that allow for an error correction and systematic improvement. As illustrated in Figure 3, this also leads to kernel parameters θ_t at different times that are independent of each other.

Below, we highlight three key properties of the PIK architecture that follow from this structure and discuss them in the subsequent sections.

(i) *Independent kernels*: PIKs convert the global task of finding a precise invertible map from the base distribution $p_0(t)$ to the final distribution $p_1(t)$ within a deep network into independent tasks of finding infinitesimal maps that solve for $\dot\phi_t(\phi; F_t)$ at each time step. For the pair $(S_t, \dot\phi_t)$ this amounts to solving (14) for $\dot\phi_t$.

(ii) *OOD resolution*: Equations (6) and (14) and respective equations for F_t provide us with an error control of this map. They are also at the root of systematic improvements in the sampling process beyond the initial training time. In particular, curing the OOD problem with the PIRG pair $(S_t, \dot\phi_t)$ boils down to locally improving the solution $\dot\phi_t(\phi)$ to a known linear differential equation, whose coefficients are determined by $F_t(\phi)$.

(iii) *Optimisation*: The properties (i) and (ii) are relevant for the setup and solution of PIKs for a given pair $(F_t, \dot\phi_t)$. The independent and analytically given structure of (14) offers further optimisation freedom for the pair, given external constraints. In particular, this includes a new kind of parameter conditional flow, systematic ways to improve inaccurate solutions of (14), and the choice of the used generating function F_t. This is discussed in more detail in Section III C.

This closes the derivation and discussion of the general PIK architecture based on PIRG pairs $(F_t, \dot\phi_t)$. In the following we concentrate on the example pair $(S_t, \dot\phi_t)$ and discuss strategies to parametrise and optimise $S_t(\phi)$ and $\dot\phi_t(\phi)$. The respective findings also apply to general pairs.

B. Sampling with the PIRG pair

Property (i) *Independent kernels* leaves us with the task of devising solution strategies for single sampling steps at times t with a pair $\big(F_t(\phi), \dot\phi_t(\phi)\big)$. The general case involves combinations of given F_t and/or $\dot\phi_t$ and this generality is specifically important for the property (iii) *Optimisation* of the PIK architecture. Here, we concentrate on solution strategies of (14) with fixed S_t.

1. Parametrisation of $S_t(\phi)$

In this case, we must choose a suitable parametrisation of $S_t(\phi)$ and following (1), we choose

$$S_t(\phi) = \hat S_t(\phi) + \log \mathcal{N}_t \,. \tag{16a}$$

Here, $\hat S_t(\phi)$ contains all field-dependent parts and $\log \mathcal{N}_t$ comprises a field-independent contribution that ensures the normalisation. We parametrise

$$\hat S_t(\phi) = \sum_i c_{i,t}\, \mathcal{O}_i(\phi)\,, \tag{16b}$$

with scale-dependent coefficients $c_{i,t}$ and a set of analytically known basis functions $\mathcal{O}_{i,t}(\phi)$. The basis functions can be chosen t-independently but in specific cases t-dependent ones may be advantageous for covering non-linear transformations within a smaller set of operators, see e.g. [27–29]. At the initial and final scale, $\{c_i\,,\, \mathcal{O}_i\}$ are chosen such that (16a) with (16b) matches the known base and target negative log likelihoods S_0, S_1. In between, the only constraint for the trajectory is its differentiability, and that (16) allows for an analytical computation of $\partial S_t(\phi)/\partial\phi$. However, secondary constraints can be used for the optimisation of the sampling process. Furthermore, this also opens the opportunity to learn optimal base distributions within the PIK architecture.

The final important ingredient in the PIK architecture is the normalisation of the flowing distribution p_t, also present in the normalisation flow architecture. For PIKs the flow of the normalisation $\mathcal{N}_t$ is directly related to that of S_t: with (5) and (16) the change of the normalisation $\mathcal{N}_t$ can be computed from that of $\hat S_t$,

$$\frac{d\log\mathcal{N}_t}{dt} = -\int \mathcal{D}\phi\, p_t(\phi) \frac{d\hat S_t(\phi)}{dt}\,. \tag{17}$$

The explicit resolution of (17) requires the computation of an expectation value for each t from a non-trivial distribution $p_t(\phi)$. This would inflict large computational costs and in Section III C we show how this task can be converted into one of solving differences of Wegner equations, see the discussion around (20).

We close the discussion of the parametrisation of $S_t(\phi)$ with iterating the remark, that the analytic parametrisation of S_t is but one of many possibilities. We have introduced it here and will use it later in the explicit examples in Section IV for its concise and read-to-use structure.

2. *Parametrisation of* $\dot{\phi}_t(\phi)$

The above parametrisation of S_t fixes all S_t-dependent terms in the Wegner equation (14), and leaves us with the task of solving it for the physics-informed kernel $\dot{\phi}_t(\phi)$. To solve (14) in practice, most known approaches require a parametrisation of the kernel $\dot{\phi}_t(\phi)$ in some set of basis functions $K_{j,t}(\phi)$,

$$\dot{\phi}_t(\phi) = \sum_j k_{j,t}\, K_{j,t}(\phi)\,, \tag{18}$$

with a chosen complete basis $\{K_{j,t}(\phi)\}$ and expansion coefficients $k_{j,t}$. With (14), this reduces the complicated optimisation problem of finding a transformation between two distributions into the problem of solving an independent set of linear systems of the form

$$A_t\, k_t = b_t\,. \tag{19}$$

Here, the matrix A_t and vector b_t are directly determined by the chosen basis $K_{j,t}(\phi)$ and the parametrised $S_t(\phi)$. Note, that because (19) is linear, finding the expansion coefficients $k_{j,t}$ directly avoids problems related to local minima.

C. Learning and optimisation

The optimisation of the PIK architecture amounts to optimising the global solution for $\dot{\phi}_t$, that is that of (19) for all t. In this process we aim for both the most rapid convergence of the given expansion and the stability of the solution for $\dot{\phi}_t$. This is in one-to-one correspondence with, firstly, a local and global optimisation of the basis $K_{j,t}(\phi)$ and, secondly, an optimisation of the determination of the respective coefficients $k_{j,t}$. This local optimisation of the basis and the coefficients is addressed in Section III C 1. Moreover, the global optimisation of the pair $(F_t, \dot{\phi}_t)$ is addressed in Section III C 2.

1. *Local optimisation*

In the PIK architecture, the problem of finding maps between two distributions is reduced to solving (14) or rather (19) for each layer. This entails that the local optimisation task in PIKs does not require machine learning. Practically speaking, the solution of (14) and (19) thrives on any successful method to solve high-dimensional linear PDEs; this includes traditional as well as machine learning-based methods. Accordingly, PIKs may immensely benefit from the extensive optimisation framework to improve traditional methods as well as neural network-based approaches to solve high-dimensional PDEs such as apt physics-informed neural networks or operators [30–32]. Moreover, recent approaches to learn optimal basis functions for PDE solvers may significantly improve the solutions computed with PIKs [33].

In practice, both traditional and machine learning-based solution strategies have to deal with the fact that (14) depends on the change of the normalisation which is not known analytically: $d\log\mathcal{N}_t/dt$ in (17) is only determined by an expectation value. However, we can use the fact that $d\log\mathcal{N}_t/dt$ is a field-independent constant to evade its explicit computation. To that end, we consider differences of Wegner equations (14) evaluated at the targeted field configuration ϕ and at some reference configuration χ. Since $d\log\mathcal{N}_t/dt$ is field-independent, it drops out in the difference. Thereby we circumvent the explicit computation or equivalently learning of $d\log\mathcal{N}_t/dt$ as done in related approaches [6, 8]. This leaves us with an efficient condition that holds for any allowed pair $(S_t, \dot{\phi}_t)$,

$$\begin{aligned} 0 = \frac{d\hat{S}_t(\phi)}{dt} - \frac{d\hat{S}_t(\chi)}{dt} \\ - \left[\frac{\partial}{\partial\phi} - \frac{\partial\hat{S}_t(\phi)}{\partial\phi}\right] \dot{\phi}_t(\phi) \\ + \left[\frac{\partial}{\partial\phi} - \frac{\partial\hat{S}_t(\phi)}{\partial\phi}\right] \dot{\phi}_t(\phi)\Big|_{\phi=\chi}\,. \end{aligned} \tag{20}$$

Crucially, this condition only requires terms that are analytically known or efficiently computable.

2. *Global optimisation*

As mentioned before, besides optimising the solution process of (14), the PIK architecture also allows to optimise the pair $(F_t, \dot{\phi}_t)$ with respect to some secondary criterion. Here, we aim to highlight a list of said criteria and comment on how they may be implemented.

Firstly, in many applications one is not merely interested in sampling from one specific distribution, but rather a family of distributions that depend on some external parameters [34–39]. To that end, parameter-conditional generative models were devised. The majority of these models work by enlarging the generative model and learning the additional parameter dependence. The parameter conditional simulations are then conducted by multiple forward passes through the enlarged network. However, for PIKs we have a direct control over the distributions seen during one forward pass, making the additional simulations and enlarged network superfluous. This will be discussed in more detail in Section IV A 3. Note that in principle this procedure could also be adapted for stochastic normalising flows and non-equilibrium transport sampler [8, 40].

Secondly, while the parameter-conditional mode of PIKs is an interesting choice, it requires a precise solution of each intermediate kernel $\dot{\phi}_t$ during the flow. However, if one is only interested in the final distribution, one can optimise the pair $(F_t, \dot{\phi}_t)$ in different ways. For instance, one may absorb the inaccuracies of the intermediate kernels $\dot{\phi}_t$ into the path F_t, requiring only a precise solution for the final kernel. Because we have analytical access to (20), we can do so in a controlled and possibly iterative manner.

This goes hand in hand with optimising the base distribution $p_{t=0}$. The latter offers a lot of potential for optimisation, as its only requirement is that one should be able to sample from it efficiently [41]. Moreover, sampling from the best base distribution for a given transformation is known from trivialising flows [16, 42–44]. The *independent kernel* property and (14) may allow one to determine closed forms for these base distributions for a given transformation, that were not within reach before.

Thirdly, while most comments above were made with the action (or negative log likelihood) in mind, we readily add that PIKs can also offer extensions to other generating functions F_t such as the rate function that store the information of the distribution in a more condensed manner. This is part of future work.

Lastly, the above comments only related to optimising the pair $(F_t, \dot{\phi}_t)$ for generative tasks. However, PIKs can, in principle, also be used to infer analytical forms of the action or Hamiltonian from a given data set. This allows to interpret the physics underlying the data set and connects PIKs to symbolic regression tasks. Practically, it is analogous to approaches based on contrastive divergence [45], where one considers whether a Langevin process with the drift of the action/Hamiltonian in question moves away from the empirical distribution given by the data. Similar to approaches based on Stein's identity [20, 46], with PIKs one can replace the Langevin process with the flow induced by the kernel $\dot{\phi}_t$. Then, optimising the kernel such that it corresponds to a trivial transformation allows to infer and optimise the analytical form of the action belonging to the data set.

This closes the derivation of the PIK architecture and its general solution procedure. In the following sections we illustrate a concrete implementation and its strengths at the example of scalar lattice field theories.

IV. PIKS AT WORK

We proceed with selected applications of the PIK architecture to tailor-made examples: scalar field theories on the lattice. These examples allow us to illustrate many of the advantageous properties of PIKs in a well-controlled setting. This Section is meant to practically illustrate the considerations made in Sections I and III. In Section I, we considered two OOD problems for generative models.

The first OOD problem relates to the size of the system: Transformations computed for a high-dimensional problem are typically inaccurate as they must interpolate in a high-dimensional space. The second OOD problem relates to the size of the training data: Transformations computed on a limited data set cannot generally be expected to have obtained knowledge on cumulants that are not sufficiently represented in the training data.

In Section IV A, we show how PIKs systematically resolve the second OOD problem: Since PIKs transform the generative task into the task of solving a sequence of independent, linear PDEs and solutions to linear PDEs can be systematically improved, PIKs inherit the same property. Accordingly, one can provide PIKs with information that goes beyond the initial training data in a controlled manner.

To that end, we begin by showing how one can construct PIKs that benefit from their (i) *independent kernels* at the example of a zero-dimensional ϕ^4-theory. Then, we explicitly comment on PIKs (ii) *OOD resolution* by correcting the PIK sampling process in a controlled way after its initial setup. Moreover, we illustrate how one can (iii) *optimise* the PIRG pair $(S_t, \dot{\phi}_t)$ at the example of a parameter-conditional flow.

In Section IV B we illustrate the applicability of PIKs to higher-dimensional probability measures. We discuss, how the respective first OOD problem, the curse of dimensionality, reduces to that of solving high-dimensional analytically known linear PDEs whose form and sparseness can be optimised to the task at hand. The resolution of this OOD task is left to a forthcoming publication.

A. Zero-dimensional ϕ^4-theory

As a first benchmark, we consider a one dimensional probability distribution $p(\varphi) = \exp\{-S(\varphi)\}$ with $\varphi \in \mathbb{R}$ in (1). In this simple case, (14) reduces to an ordinary differential equation. This example can be understood as a zero-dimensional scalar field theory and we restrict ourselves to a ϕ^4-theory with the action

$$\hat{S}(\varphi) = \frac{1}{2} m^2 \varphi^2 + \frac{\lambda}{4!} \varphi^4 \,, \qquad \varphi \in \mathbb{R} \,, \tag{21}$$

with the two parameters m^2 (mass squared) and λ (coupling). This system can also be solved by direct integration which allows for a good comparison.

For the resolution of the sampling problem with PIKs we define a PIRG pair with an action S_t, that is a linear interpolation between the base action S_0 and the target action S_t, similar to [40]:

$$\hat{S}_t(\phi) = \frac{1}{2} m^2(t)\, \phi^2 + \frac{\lambda(t)}{4}\, \phi^4 \,, \qquad \phi \in \mathbb{R} \,, \tag{22a}$$

with

$$\begin{aligned} m^2(t) &= m_0^2 + t\, \left(m_1^2 - m_0^2\right) \\ \lambda(t) &= \lambda_0 + t\, \left(\lambda_1 - \lambda_0\right) \,. \end{aligned} \tag{22b}$$

Here $(m_0^2\,, \lambda_0)$ are the parameters of the base distributions and $(m_1^2\,, \lambda_1)$ are those of the target distribution. Note that this parametrisation is a free choice and naturally lends itself to parameter-conditional models as will be discussed in Section IV A 3. This is a specific kind of optimisation and other examples can be found in Section III C.

For this toy model, we use a flow from a unimodal Gaussian theory with

$$m_0^2 = 1\,, \quad \lambda_0 = 0\,, \tag{23}$$

to a theory with a multimodal behaviour and $\lambda \neq 0$,

$$m_1^2 = -2\,, \quad \lambda_1 = \frac{1}{2}\,. \tag{24}$$

Note that capturing multimodal target distributions poses a difficulty for many traditional generative models because of the phenomenon of mode-collapse or mode-seeking [47–49]. PIKs circumvent this problem by enforcing a particular path S_t, which fixes the way the unimodal distribution is morphed into the multimodal one. Moreover, the independent kernels $\dot\phi_t$ allow for an error correction at each time step.

1. *Independent kernels* $\dot\phi_t$

During the flow, the field ϕ_t is propagated in time according to (11). This ODE in time is solved numerically by a standard fourth order Runge-Kutta scheme with a step size of $\Delta t = 1/25$, leaving us with $N_t = 51$ kernels $\dot\phi_t$ to be determined. Because we have fixed the path $S_t(\phi)$ via (22a) and (22b), each kernel has its own independent Wegner equation (14) it must solve.

In analogy to the parametrisation in (18), we express each kernel $\dot\phi_t(\phi)$ in a chosen basis following a similar approach as in [37],

$$K_{1,t}(\phi) = \phi\,, \quad K_{j,t}(\phi) = \sin\left(\omega_{j,t}\,\phi\right) \quad j \geq 2\,. \tag{25}$$

We choose this basis because $\hat{S}_t(\phi)$ is symmetric under the transformation $\phi \to -\phi$. To ensure that the distribution generated by the PIKs carries the same symmetry, we want $\dot\phi_t$ to transform equivariantly under this sign flip [50].

In analogy to other generative models, we use stochastic gradient descent with the Adam optimiser to compute the coefficients $k_{j,t}$ and frequencies $\omega_{j,t}$. Furthermore, we use the mean squared error induced by (20) as a loss function. Here, the fact that the change in normalisation $d \log \mathcal{N}_t/dt$ does not have to be computed is used explicitly. We choose $\chi = 0$ as a reference configuration for (20), which leads to a vanishing time and field derivative of the action in (20).

The loss is bounded from above by 10^{-4} after saturation. Accordingly, the effective sample size also effectively remains at unity. More details of the training are provided in Appendix A.

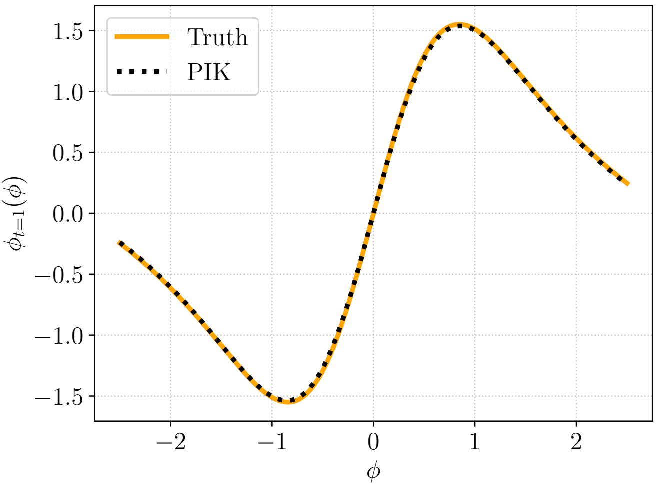

Figure 4. Comparison of the true (orange) and the learned (black) kernel $\dot\phi_{t=1}(\phi)$ for the zero-dimensional ϕ^4-theory.

In order to show the successful training of the kernel, we compare the learned kernel $\dot\phi_t(\phi)$ at $t = 1$ to the one obtained by directly integrating (14) in Figure 4 and we find a very good overlap.

Figure 5 shows a comparison of the targeted (orange) and modelled (black) distribution at the final time $t = 1$. We find that the well-trained PIK transformation indeed connects the initial Gaussian distribution to the targeted multimodal one. The target distribution was computed using 10^6 samples from a standard hybrid Monte Carlo algorithm (HMC). The modelled distribution was generated by pushing 10^6 samples from the initial Gaussian distribution through the PIK flow. We find an excellent agreement between both distributions.

We close the discussion of the independent kernels $\dot\phi_t$ with a hands-on assessment of the speedup in runtime gained by choosing PIKs over HMC. Evidently, the performance of both HMC and PIKs depends crucially on the specific implementation [51], and we refrain from optimising neither HMC nor PIKs for speed. Such a comprehensive analysis goes beyond the scope of the present work, and hence the present comparison only offers a rough estimate for the speedup. Specifically, for HMC we use the standard leapfrog integrator with the same discretisation in time as for PIKs and integrate trajectories up to a final time $T = 1$. The number of effectively independent samples generated by HMC is computed using the Γ-method [52]. With this setup, one generates the same number of independent samples with PIKs around 20 times faster compared to HMC. While only being a rough estimate, we consider such a speedup very promising.

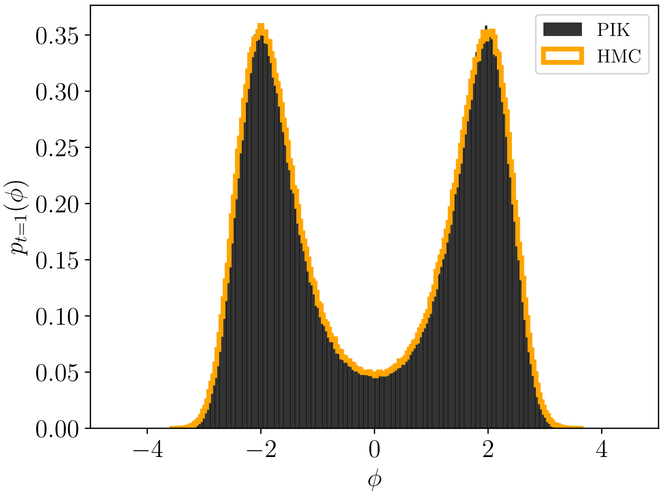

Figure 5. Comparison of the targeted (orange, HMC) and modelled (black, PIK) zero-dimensional ϕ^4 distribution.

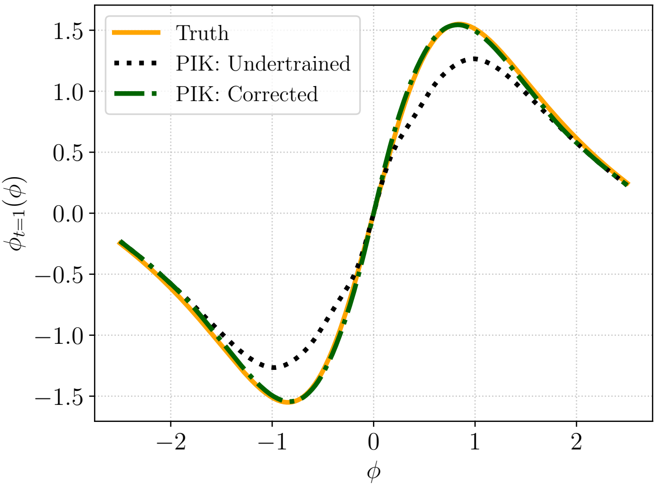

Figure 6. Comparison of the true (orange), the undertrained (black), and the corrected (green) kernel $\dot{\phi}_t(\phi)$ at $t = 1$ for the zero-dimensional ϕ^4-theory.

2. *OOD resolution*

Having showcased that PIKs have the capacity to learn each kernel $\dot{\phi}_t$ independently without the requirement of computing the change in the normalisation, we now turn to the property (ii) *OOD resolution*. Instead of the explicit volume scaling, here we illustrate the systematic and controlled improvability of the PIK sampling process after its initial setup and training. This is naturally enabled by the fact that we have analytical access to the Wegner equation (14) (or (20), respectively) that has to be satisfied by each kernel individually. It is further boosted by the fact that the Wegner equation is linear in $\dot{\phi}_t$, making the improvements more directly accessible.

As discussed in Section I, already the necessarily limited training data often leads to OOD problems for generative models. Here, one can use additional Monte Carlo steps to cure these problems [40, 53]. However, since this can become costly and does not improve the inherent quality of the model, one may look for more optimal solutions. With PIKs, we can directly use the information of the distribution via $S_t(\phi)$ at each time step to systematically improve the sampling beyond training time.

In order to mimic this OOD problem, we purposefully undertrain the last layer of the PIK flow, such that the kernel only saw $\sim 50\,k$ samples during the optimisation. This undertrained or approximate kernel is referred to as $\dot{\phi}_t^{(0)}$. Note that also for other continuous generative models, the last layer must exactly solve (20) in order to be truly generative. The undertrained kernel together with the true solution is depicted in Figure 6 in black and orange, respectively. Here, a clear deviation between the two is visible.

In order to systematically improve the sampling, we must enrich the solution of the kernel $\dot{\phi}_t^{(0)}$ such that it better satisfies (20). There are manifold ways to achieve this in a controlled way [33, 54–58]. Here, we use the linear structure of (20). For notational convenience, we denote the second and third line of (20) as

$$\mathcal{L}_\chi\, \dot{\phi}_t(\phi) := \left[\frac{\partial}{\partial\phi} - \frac{\partial \hat{S}_t(\phi)}{\partial\phi}\right] \dot{\phi}_t(\phi) \\ - \left[\frac{\partial \hat{S}_t(\phi)}{\partial\phi} - \frac{\partial}{\partial\phi}\right] \dot{\phi}_t(\phi)\Big|_{\phi=\chi}\,, \tag{26}$$

defining the linear operator $\mathcal{L}_\chi$ with the reference configuration χ. The residual of the approximate kernel $\dot{\phi}_t^{(0)}$ is then readily computed as

$$r_\chi^{(0)}(\phi) = \frac{d\hat{S}_t(\phi)}{dt} - \frac{d\hat{S}_t(\chi)}{dt} - \mathcal{L}_\chi\, \dot{\phi}_t^{(0)}(\phi)\,. \tag{27}$$

Naturally, a correction $\Delta\dot{\phi}_t^{(0)}(\phi)$ that satisfies

$$\mathcal{L}_\chi\, \Delta\dot{\phi}_t^{(0)}(\phi) = r_\chi^{(0)}(\phi)\,, \tag{28}$$

will lead to a kernel

$$\dot{\phi}_t^{(1)}(\phi) = \dot{\phi}_t^{(0)}(\phi) + \Delta\dot{\phi}_t^{(0)}(\phi)\,, \tag{29}$$

that solves (20) accurately. As is already hinted at by the superscript notation, this procedure can in principle be iterated because of the linear structure of the Wegner equation, allowing for controlled corrections [58].

In order not to spoil the well-fitted regions of the undertrained kernel and because they more readily transfer to higher-dimensional problems, we solve (28) with a collocation approach using compactly supported radial basis functions, more concretely Wendland functions [59–61]. To ensure the $\mathbb{Z}_2$ symmetry of the transformation, the basis functions were anti-symmetrised. The centres

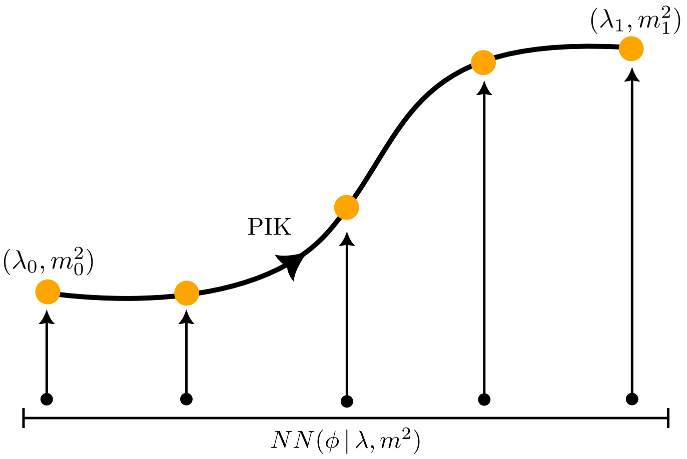

Figure 7. Difference in the kind of parameter conditionality for PIKs differs and traditional generative models $NN(\phi \,|\, \lambda, m^2)$. While PIKs move along the path determined by the parameters, traditional models need to sample each set of configurations individually.

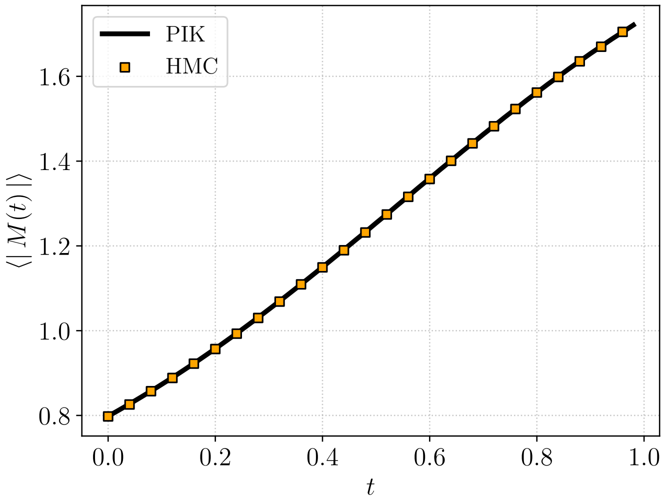

Figure 8. Magnetisation of the zero-dimensional ϕ^4-theory computed with PIKs in a continuous manner (black) and HMC with individual simulations (orange).

of the radial basis functions were chosen from regions where the undertrained kernel deviated most from the true one. Here, we used eleven radial basis functions. Because the latter are compactly supported, the resultant linear system similar to (18) is sparse and readily solved. Further details of the correction procedure are provided in Appendix B.

In Figure 6, we show the corrected kernel $\dot{\phi}_t^{(1)}(\phi)$ in green together with the undertrained and true kernel in black and orange, respectively. We find that the correction indeed leads to a kernel that is very close to the true one and corrects the undertrained one in the regions where it deviates the most.

While the correction mechanisms can be readily improved, this example illustrates the general systematic workflow of PIKs from (a) an (imprecise) global solution to (b) a determination of OOD regions during sampling or using information of the residual to (c) a controlled (and local) correction of the kernel. This is natural for PDE approaches [58] but the proposed method is to our knowledge novel in the context of generative models.

3. *Optimisation with parameter-conditional PIKs*

The above sections showcased the setup, training, and systematic improvement of PIKs. In this section, we comment on the property *(iii) Optimisation* of PIKs. As outlined in Section III C, there are manifold ways to optimise the pair $(S_t, \dot{\phi}_t)$ in order to improve the sampling process, given auxiliary constraints. Here, we illustrate one such optimisation at the example of a parameter-conditional flow.

With PIKs we have direct control over the concrete distribution at each time step. Here, we use this freedom to enforce the same functional form of the action at each time step. The only thing that changes for the distribution in the flow are the parameters (m^2, λ) at each time step. This allows us to make PIKs parameter conditional without inflating the network.

Unless one salvages a specific structure of the action [34], parameter-conditional networks $NN(\phi \,|\lambda, m^2)$ commonly work by enlarging the network in a constructive way such that it can learn the additional parameter-dependence [37, 38]. For PIKs, the flow itself can be made parameter-conditional as described above such that any additional structure and learning becomes superfluous. This is also illustrated in Figure 7. Moreover, this directly allows us to compute observables in a continuous way without sampling the configurations for each parameter individually.

We exemplify the PIK approach in Figure 8 for the observable

$$\langle |M(t)| \rangle = \int \mathcal{D}\phi \; p_t(\phi) \, |\phi| \,, \tag{30}$$

which corresponds to the expected magnetisation for the zero-dimensional ϕ^4-theory. Here we draw a batch of samples from the Gaussian theory at $t = 0$ and push it through the PIK flow. The observable (black) can then be computed at each intermediate time step and is purposefully depicted with a continuous line. For comparison, we also show the same observable computed with individual HMC simulations, where each point corresponds to a separate simulation.

B. Higher-dimensional ϕ^4-theories

In the previous Section IV A, we have used the zero-dimensional ϕ^4-theory to illustrate and benchmark PIKs in an analytically accessible example. In particular, we have showcased how PIKs can evade the OOD problem pertaining to the limited information in the training data. The remaining OOD problem, the curse of dimensionality, is only present in higher-dimensional systems and its proposed resolution is discussed here. In Section IV B 1 we discuss the general application of PIKs to higher-dimensional systems. In Section IV B 2 we detail how PIKs can be used to systematically tackle the curse of dimensionality. A full numerical study goes beyond the scope of the present work and is the subject of a forthcoming publication.

As an exemplary system, we consider a d-dimensional lattice ϕ^4-theory with V sites and the action

$$\hat{S}_t(\phi) = \sum_{\hat{n}\in\mathcal{D}} \Bigg[\frac{1}{2}\left(m^2(t) + 2d\right)\phi_{\hat{n}}^2 - \sum_{\mu=1}^{d} \phi_{\hat{n}}\,\phi_{\hat{n}+\hat{\mu}} + \frac{\lambda(t)}{4}\phi_{\hat{n}}^4 \Bigg] . \tag{31}$$

Here, the coupling $\lambda(t)$ and squared mass $m^2(t)$ are linear in t and are given by (22b). Moreover, $\hat{\mu}$ is the unit vector in the direction μ with $\mu = 1, ..., d$. Specifically interesting corners in these higher dimensional theories are the regimes that host second order phase transitions. There, standard algorithms face critical slowing down as the field values on well-separated lattice sites become strongly correlated which makes traditional sampling algorithms such as HMC inefficient [62]. Moreover, also known generative models struggle to learn adequate transformations in this regime [63, 64]. Below we discuss how PIKs can help to tackle these problems for generative models.

1. *PIK setup for high-dimensional theories*

High-dimensional systems introduce additional redundancies, most importantly symmetries, that leave the action invariant. For generative models, these redundancies can be treated with equivariant flows in order to improve the sampling quality [37, 50, 65, 66]. In the following, we show that also PIKs are able to easily and efficiently eliminate these redundancies. In doing so, we further show how solving for the kernel can be reduced to solving for a single function Ω_t instead of a vector field $\dot{\phi}_t$ directly, see also [8, 50, 67]. This is accompanied by a discussion of boundary conditions, which implement the normalisation of the distribution and define a unique solution of the kernel.

As mentioned above, PIKs as well as standard generative approaches optimise a high-dimensional vector field, which entails optimising V functions. This process can be simplified by writing the kernel as a gradient field [8, 50, 67] and in the context of PIKs the kernel in the Wegner equation (14) admits a Helmholtz decomposition. Thus, we can express the kernel in terms of a function $\Omega_t(\phi)$ and a vector field $H_t(\phi)$

$$\dot{\phi}_t(\phi) = -\frac{\partial}{\partial\phi}\,\Omega_t(\phi) + H_t(\phi)\,, \qquad 0 = \frac{\partial}{\partial\phi}\left[p_t(\phi)\,H_t(\phi)\right] . \tag{32}$$

For the Wegner equation, this directly manifests as

$$\frac{dS_t(\phi)}{dt} = -\left[\frac{\partial}{\partial\phi} - \frac{\partial S_t(\phi)}{\partial\phi}\right]\frac{\partial}{\partial\phi}\,\Omega_t(\phi) + \underbrace{\left[\frac{\partial}{\partial\phi} - \frac{\partial S_t(\phi)}{\partial\phi}\right] H_t(\phi)}_{=0} . \tag{33}$$

Accordingly, the additional vector field $H_t(\phi)$ does not contribute to the change in the action. Rather, it describes permutations and rotations in the field space that are irrelevant for the change in the measure. Typically, one would like to discard those to improve the sampling quality [50] and it is a nice feature of PIKs that it does so naturally. With this, rather than optimising and parametrising V functions, one must merely solve the adapted Wegner equation

$$\left[\frac{\partial^2}{\partial\phi^2} - \frac{\partial S_t(\phi)}{\partial\phi}\,\frac{\partial}{\partial\phi}\right]\Omega_t(\phi) = -\frac{dS_t(\phi)}{dt}\,, \tag{34}$$

for one function $\Omega_t(\phi)$. Here, $\partial^2/\partial\phi^2$ denotes the canonical Laplace operator. Note that (34) is a linear PDE of elliptic type. Accordingly, it benefits from the insights gained in Section IV A.

To determine the suitable boundary conditions, we integrate the Wegner equation (14) over a hypercube $\mathcal{V}$ with radius R in all directions. Using the divergence theorem, this straightforwardly leads to

$$-\int_{\mathcal{V}} \mathcal{D}\phi\, p_t(\phi)\,\frac{d\hat{S}_t(\phi)}{dt} - \frac{d\log\mathcal{N}_t}{dt} = \int_{\partial\mathcal{V}} \mathcal{D}\phi_s\, p_t(\phi_s)\,\dot{\phi}_t(\phi_s)\,. \tag{35}$$

Here, the right-hand side denotes a surface integral over the boundary $\partial\mathcal{V}$ of the hypercube and $\mathcal{D}\phi_s$ refers to the respective measure. Letting $R\to\infty$, one may use the definition of $\log\mathcal{N}_t$ and (17) to see that the left-hand side vanishes. This leads to the boundary condition

$$\lim_{R\to\infty}\int_{\partial\mathcal{V}} \mathcal{D}\phi_s\, p_t(\phi_s)\,\dot{\phi}_t(\phi_s) = 0\,, \tag{36}$$

for the kernel and ensures the normalisation of the distribution during the flow.

In summary, (34) together with (32) and (36) encodes a complete, redundancy-reduced version of the propagation step for the generative task with the path S_t.

2. Tackling the curse of dimensionality

In standard black-box approaches to the generative task, reducing the complexity of finding an accurate vector field is typically limited to salvaging known symmetries of the theory [37, 50, 68]. In PIKs however, the curse of dimensionality can be tackled by using the analytic structure at hand to reduce the complexity of the system: the adapted Wegner equation (34) provides us with insights about a good basis, leading to a more rapid convergence to the solution. For example, in the non-interacting case ($\lambda = 0$) the Eigenfunctions of the operator on the left-hand side of (34) are properly chosen tensor products of Hermite polynomials. Since Hermite polynomials can be used to form a complete basis, they are well-suited to expand Ω_t. Compared to black-box approaches, this is assumed to reduce the number of required optimisable parameters.

For the present example (31), fully optimised PIKs may allow to limit the number of required basis elements to only grow with the spacetime dimension d rather than the volume V. To that end, we note that the right-hand side of (34) only includes terms $\phi^2_{\hat{n}}, \phi^4_{\hat{n}}$ that do not couple different lattice sites, see (31). Moreover, because of the local structure of S_t, the differential operator in (34) maps tensor product of nearest neighbours back to tensor products of nearest neighbours. However, tensor products that include more distant lattice sites generate new terms that need to be cancelled again as they do not appear on the right-hand side. Note that different to CNNs, this parameter reduction does not only follow from the translational invariance of the system but rather from the local structure of the action and the Wegner equation, which is a stronger statement.

We may also utilise the property of the PIK, that the kernel transformation implicitly also includes a standard rescaling RG step. This property becomes even explicit if changing the generating function $F_t = S_t$ to the Wilson action or the effective action. This is analogous to the inverse RG as used in [41, 69–72] and will be discussed in a forthcoming work.

Finally we can optimise the adapted Wegner equation (34) by varying the path S_t. This changes the S_t-dependent coefficients of (34), see Section III C 2. Optimised choices for the coefficients simplify the task of solving (34).

In summary, the solution of fully optimised PIKs may scale with the spacetime dimension as discussed above. This would reduce a problem with a naïve scaling with the volume to one that only scales with the spacetime dimension. Already without using this additional structure, promising results have been obtained in the related approach [8] within an application to a two-dimensional ϕ^4-theory. Here, the PIK setup allows us to additionally exploit the structure provided by the Wegner equation. The remaining challenge of ensuring the accuracy of the solution to the Wegner equation for PIKs is then reduced to an OOD problem of the second type whose solution was already illustrated in Section IV A 2.

V. SUMMARY AND OUTLOOK

In this work, we introduced physics-informed kernels (PIKs) as a new architecture for generative models. The PIK-architecture aims at solving OOD problems in generative models as well as implementing a systematic error control. At its heart, PIKs reduce the task of finding transformations between two distributions into a decoupled chain of layerwise transformations that are given by the analytic continuity equations. For the distribution itself the continuity equation is the Wegner equation (4), for the rate function the continuity equation is the generalised flow equation (6), and similar ones for other generating functions. Viewed as PDEs for the kernels these equations boil down to independent linear differential equations whose solution can be systematically corrected. As a paradigmatic example, we have considered PIKs for the distribution with completely determined action paths $S_t(\phi)$ that lead to simple Wegner equations.

In general, PIKs thrive in situations where one has some access to the action, Hamiltonian, or negative log likelihood of the system at hand. In comparison to diffusion models, flow matching, and related architectures [17–19, 21–26, 72], PIKs offer direct ways to implement this knowledge into the training and sampling without requiring Monte Carlo steps or simulations. In short, PIKs aim to use this property to systematically tackle OOD problems in generative architectures.

In the present work we have shown how one can set up PIKs with (i) *independent Kernels*. Moreover, we illustrated that one can implement an error correction to systematically approach a (ii) *OOD resolution*. Lastly, we illustrated that one can use the gained freedom of PIKs to (iii) *optimise* the PIRG pair $(S_t, \dot{\phi}_t)$ at the example of a new form of parameter-conditional generative models.

Thus, this work paves the way for many interesting applications and improvements of PIKs highlighted throughout the text.

The most prominent direction is the application of PIKs to currently studied theories in physics and related fields. This avenue includes the application to higher-dimensional theories as detailed in Section IV B. It further includes incorporating gauge and fermionic degrees of freedom into the flow. Since the respectively adapted Wegner equation is known, PIKs can be straightforwardly applied to these systems. In these applications to currently relevant theories, PIKs explicitly target OOD problems encountered for generative models and are therefore an interesting target to show the practical advantages of generative sampling.

Supporting these developments opens up further avenues for future research by the manifold opportunities for both local and global optimisation as detailed in Section III C. PIKs dissect the generative task into solving sets of linear differential equations. Systematically improving (parts of) this solution process is an important technical task ideally suit for machine learning applications. This includes salvaging PINNs, neural operators, optimised basis expansions, or improved learned paths for the action to solve the respective equations more efficiently. Targeting the global optimisation, opens up the possibility to learn optimal base distributions, benefit from improved generating functions F_t, or infer the action for a given data set, see Section III C 2.

In combination, PIKs form a generative model that splits the generative task into well-controlled subproblems given by analytically known and independent differential equations determined by the underlying physics. This structure allows for systematic corrections, optimisations of individual parts of the algorithm, and thereby targets OOD problems in generative models.

ACKNOWLEDGMENTS

We thank Gert Aarts, Kenji Fukushima, Sander Hummerich, Thore Kolja Joswig, Ullrich Köthe, Timoteo Lee, Manfred Salmhofer, Robert Scheichl, Lingxiao Wang and Kai Zhou for discussions and collaborations on related subjects. FI and RK would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge for support and hospitality during the programme RCLW04, supported by the EPSRC grant no EP/Z000580/1. We acknowledge the DM-LFT Collaboration for stimulating discussions and continuous support throughout this work. It is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster). We also acknowledge support by the state of Baden-Württemberg through bwHPC.

Appendix A: Training details

In Section IV A 1, we have discussed how one can compute the independent kernel using standard machine learning techniques. In this Appendix we expand on the details of the present implementation. As described in the introduction, at each time-step t an independent kernel is computed using the samples ϕ_t, which then provides the new $\phi_{t+\Delta t}$. Samples for the training of the kernels are drawn from the base distribution defined by S_0 and successively evolved along the RG-time t, by pushing them through previously computed kernels. Accordingly, for the training only samples from the base distributions are required.

The independent kernels are parametrised by (18) and basis functions are given by (25), using a linear term and sine-functions with learnable frequencies $\omega_{j,t}$. The respective coefficients and frequencies are computed using the Adam optimiser, optimising with respect to the mean squared error induced by (20).

To compute the RG-time evolution, we chose a time step $\Delta t = 1/25$ within the fourth order Runge-Kutta scheme, which requires the determination $N_t = 51$ kernels. For each kernel, we chose $N_f = 25$ learnable frequencies. The first two time-steps kernels were trained with a learning rate of 10^{-2} for 600 epochs using a batch size of 512 samples that were drawn from the respective initial distributions. For the subsequent kernels, we transferred the weights of the previous kernel and trained them with a learning rate of 10^{-3} for maximally 600 epochs and used early stopping w.r.t the loss function to reduce the training time. For all kernels, this training procedure achieved a mean squared error loss that was bounded from above by 10^{-4} after training, achieving an effective sample size very close to unity.

After writing, we took note that this approach of learning the kernels iteratively is similar in spirit to an approach put forward in [26], where the map is regressed from data of an unknown distribution. Note however, that PIKs differ qualitatively from this approach as they use a completely known path F_t for the generating function, transfer the generative task to solving linear differential equations, and allow for systematic corrections beyond the initial training data.

Appendix B: Correction by collocation methods

In Section IV A 2, we discussed how one can systematically correct a kernel that captures some but not all of the relevant features of the true kernel. While there are many possible mechanisms to improve the solution of a PDE [33, 54–58], here we illustrate one such method based on a collocation approach with radial basis functions [60, 61], which is a simple and straightforward choice.

As discussed in Section IV A 2, for an imperfect kernel $\dot\phi_t^{(0)}(\phi)$, one can easily compute the residual $r_\chi^{(0)}(\phi)$ w.r.t (20) and the reference configuration χ. To correct the kernel, we would like to solve the linear PDE

$$\mathcal{L}_\chi \, \Delta\dot\phi_t^{(0)}(\phi) = r_\chi^{(0)}(\phi) \,, \tag{B1}$$

for the correction term $\Delta\dot\phi_t^{(0)}(\phi)$ [58]. The linear operator $\mathcal{L}_\chi$ was defined in (26). Similar to (18), we parametrise $\Delta\dot\phi_t^{(0)}(\phi)$ as

$$\Delta\dot\phi_t^{(0)}(\phi) = \sum_{i=1}^{M} w_i \, W_i(\phi, \phi_i) \,. \tag{B2}$$

Here, $\{\phi_i\}_{i=1}^M$ denote the fixed centres of the radial basis functions. We choose them to be in regions with a large mismatch between the true and the approximate kernel. Moreover, $W_i(\phi, \phi_i)$ relates to Wendland functions centred at ϕ_i, which is a compactly supported and positive definite radial basis function [59]. This allows for the a sparse linear system and local corrections to the approximate kernel. For the computations of this paper, we chose four times differentiable Wendland functions

$$W(r) = (1-r)_+^{l+2} \left[(l^2+4l+3)r^2 + (3l+6)r + 3 \right] \,, \tag{B3}$$

with $r \in \mathbb{R}$ and $l = \lfloor d/2 \rfloor + 3$. Here, $(1-r)_+ = \max(1-r, 0)$ ensures the compact support. To ensure that $\Delta\dot{\phi}_t^{(0)}(\phi)$ is also odd, we anti-symmetrised the basis functions accordingly

$$W_i(\phi, \phi_i) = W\left(\frac{|\phi - \phi_i|}{\sigma}\right) - W\left(\frac{|\phi + \phi_i|}{\sigma}\right) . \tag{B4}$$

The shape parameter σ is tuned to ensure a good overlap of the basis functions. For the computations of this paper, we chose $\sigma \approx 0.41$. As discussed in the main text, we chose $M = 11$ centres ϕ_i that are distributed in the region where the approximate kernel deviates most from the true kernel. To determine the coefficients w_i, we gathered $N = 501$ linearly spaced collocation points in the same region and solved the resultant overdetermined linear system as in (19) using a QR decomposition.

Finally, we note the present implementation is straightforward but not a unique choice and other methods may offer a significantly better performance. The optimisation of these auxiliary methods is, however, left for future work.

Appendix C: PIKs as finite time corrections to the Langevin algorithm

In Section III, we have seen how PIKs are constructed and how they can be used as generative models. By noting that generative models at their core facilitate a very fast thermalisation time for sampling algorithms, we discuss how PIKs conceptually take the role of finite time corrections to the Langevin algorithm which thermalises only in the infinite time limit. For related discussions, see also [8]. This connects the deterministic and continuous PIK model with the stochastic differential equation of the Langevin algorithm and is indeed analogous to the connection between continuous normalising flows and diffusion models [23].

For the Langevin algorithm, one considers the stochastic process

$$\frac{d\phi_t}{dt} = f_t(\phi) + g_t\, \eta_t \,, \tag{C1}$$

where one uses the drift $f_t(\phi)$, the uncorrelated Gaussian noise η_t with the pre-factor $g_t = \sqrt{2}$. This transformation induces the change in action at each time step according to

$$\frac{dS_t(\phi)}{dt} = \frac{\partial}{\partial\phi} f_t(\phi) - f_t(\phi) \cdot \frac{\partial}{\partial\phi} S_t(\phi) + \frac{\partial^2}{\partial\phi^2} S_t(\phi) - \left[\frac{\partial}{\partial\phi} S_t(\phi)\right]^2 . \tag{C2}$$

As is known, in the infinite time limit, $t \to \infty$, one arrives at the stationary distribution $\propto e^{-S(\phi)}$ by choosing $f_t(\phi) = -\frac{\partial}{\partial\phi} S(\phi)$. However, to arrive at the same distribution at some fixed and finite time, one can introduce a finite time correction to this approach via

$$f_t(\phi) = -\frac{\partial}{\partial\phi} S_t(\phi) + \dot{\phi}_t(\phi) \,, \tag{C3}$$

where $\partial S_t(\phi)/\partial\phi$ is the drift term for the time-dependent action. Using this Ansatz in (C2), one directly retrieves the Wegner equation as used in the main body of the text

$$\frac{dS_t(\phi)}{dt} = \frac{\partial}{\partial\phi}\, \dot{\phi}_t(\phi) - \dot{\phi}_t(\phi)\, \frac{\partial}{\partial\phi} S_t(\phi) \,. \tag{C4}$$

From this point of view, one can see the PIKs as a constructive approach to accelerate the thermalisation time of the Langevin algorithm by introducing a finite time correction to the drift term.

[1] F. Ihssen and J. M. Pawlowski, Physics-informed renormalisation group flows, Annals Phys. **481**, 170177 (2025), arXiv:2409.13679 [hep-th].

[2] F. J. Wegner, Some invariance properties of the renormalization group, Journal of Physics C: Solid State Physics **7**, 2098 (1974).

[3] J. M. Pawlowski, Aspects of the functional renormalisation group, Annals Phys. **322**, 2831 (2007), arXiv:hep-th/0512261.

[4] K. Holland, A. Ipp, D. I. Müller, and U. Wenger, Machine-learned rg-improved gauge actions and classically perfect gradient flows (2025), arXiv:2504.15870 [hep-lat].

[5] J. Cotler and S. Rezchikov, Renormalization group flow as optimal transport, Phys. Rev. D **108**, 025003 (2023), arXiv:2202.11737 [hep-th].

[6] B. Máté and F. Fleuret, Learning interpolations between boltzmann densities (2023), arXiv:2301.07388 [stat.ML].

[7] J. Cotler and S. Rezchikov, Renormalizing diffusion models (2023), arXiv:2308.12355 [hep-th].

[8] M. S. Albergo and E. Vanden-Eijnden, Nets: A non-equilibrium transport sampler (2025), arXiv:2410.02711 [cs.LG].

[9] A. Sheshmani, Y.-Z. You, B. Buyukates, A. Ziashahabi, and S. Avestimehr, Renormalization group flow, optimal transport, and diffusion-based generative model, Phys. Rev. E **111**, 015304 (2025).

[10] J. Polchinski, Renormalization and effective lagrangians, Nuclear Physics B **231**, 269 (1984).

[11] C. Wetterich, Exact evolution equation for the effective potential, Phys. Lett. B **301**, 90 (1993), arXiv:1710.05815 [hep-th].

[12] U. Ellwanger, FLow equations for N point functions and bound states, Z. Phys. C **62**, 503 (1994), arXiv:hep-ph/9308260.

[13] T. R. Morris, The Exact renormalization group and approximate solutions, Int. J. Mod. Phys. A **9**, 2411 (1994),

arXiv:hep-ph/9308265.

[14] F. Ihssen and J. M. Pawlowski, Functional flows for complex effective actions, SciPost Phys. **15**, 074 (2023), arXiv:2207.10057 [hep-th].

[15] F. Ihssen and J. M. Pawlowski, Flowing fields and optimal rg-flows (2023), arXiv:2305.00816 [hep-th].

[16] M. Lüscher, Trivializing maps, the Wilson flow and the HMC algorithm, Commun. Math. Phys. **293**, 899 (2010), arXiv:0907.5491 [hep-lat].

[17] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows (2016), arXiv:1505.05770 [stat.ML].

[18] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019), arXiv:1904.12072 [hep-lat].

[19] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models, Phys. Rev. Lett. **126**, 032001 (2021), arXiv:2007.07115 [hep-lat].

[20] Q. Liu and D. Wang, Stein variational gradient descent: A general purpose bayesian inference algorithm (2019), arXiv:1608.04471 [stat.ML].

[21] L. Dinh, J. Sohl-Dickstein, and S. Bengio, Density estimation using real nvp (2017), arXiv:1605.08803 [cs.LG].

[22] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models (2020), arXiv:2006.11239 [cs.LG].

[23] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations (2021), arXiv:2011.13456 [cs.LG].

[24] L. Wang, G. Aarts, and K. Zhou, Diffusion models as stochastic quantization in lattice field theory, JHEP **05**, 060, arXiv:2309.17082 [hep-lat].

[25] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, Flow matching for generative modeling (2023), arXiv:2210.02747 [cs.LG].

[26] C. Xu, X. Cheng, and Y. Xie, Normalizing flow neural networks by jko scheme (2024), arXiv:2212.14424 [stat.ML].

[27] F. Lamprecht, Diploma thesis Heidelberg University (2007).

[28] F. Isaule, M. C. Birse, and N. R. Walet, Application of the functional renormalization group to Bose gases: from linear to hydrodynamic fluctuations, Phys. Rev. B **98**, 144502 (2018), arXiv:1806.10373 [cond-mat.quant-gas].

[29] F. Isaule, M. C. Birse, and N. R. Walet, Thermodynamics of Bose gases from functional renormalization with a hydrodynamic low-energy effective action, Annals Phys. **412**, 168006 (2020), arXiv:1902.07135 [cond-mat.quant-gas].

[30] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics **378**, 686 (2019).

[31] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar, Physics-informed neural operator for learning partial differential equations (2023), arXiv:2111.03794 [cs.LG].

[32] J. Richter-Powell, Y. Lipman, and R. T. Q. Chen, Neural conservation laws: A divergence-free perspective (2022), arXiv:2210.01741 [cs.LG].

[33] M. Weng, Z. Mao, and J. Shen, Deep collocation method: A framework for solving pdes using neural networks with error control (2025), arXiv:2502.17203 [math.NA].

[34] Q. Zhu, G. Aarts, W. Wang, K. Zhou, and L. Wang, Diffusion models for lattice gauge field simulations, in *38th conference on Neural Information Processing Systems* (2024) arXiv:2410.19602 [hep-lat].

[35] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, Learning likelihoods with conditional normalizing flows (2023), arXiv:1912.00042 [cs.LG].

[36] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, Analyzing inverse problems with invertible neural networks (2019), arXiv:1808.04730 [cs.LG].

[37] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, Learning lattice quantum field theories with equivariant continuous flows, SciPost Phys. **15**, 238 (2023), arXiv:2207.00283 [hep-lat].

[38] A. Singha, D. Chakrabarti, and V. Arora, Conditional normalizing flow for Markov chain Monte Carlo sampling in the critical region of lattice field theory, Phys. Rev. D **107**, 014512 (2023), arXiv:2207.00980 [hep-lat].

[39] S. Wahl, A. Rousselot, F. Draxler, H. Schopmans, and U. Köthe, Trade: Transfer of distributions between external conditions with normalizing flows (2025), arXiv:2410.19492 [cs.LG].

[40] H. Wu, J. Köhler, and F. Noé, Stochastic normalizing flows (2020), arXiv:2002.06707 [stat.ML].

[41] M. Bauer, R. Kapust, J. M. Pawlowski, and F. L. Temmen, Super-Resolving Normalising Flows for Lattice Field Theories, SciPost Phys. **19**, 077 (2025), arXiv:2412.12842 [hep-lat].

[42] D. Albandea, L. Del Debbio, P. Hernández, R. Kenway, J. Marsh Rossney, and A. Ramos, Learning trivializing flows, Eur. Phys. J. C **83**, 676 (2023), arXiv:2302.08408 [hep-lat].

[43] D. Albandea, L. Del Debbio, P. Hernández, R. Kenway, J. M. Rossney, and A. Ramos, Learning Trivializing Flows in a ϕ^4 theory from coarser lattices, PoS **LATTICE2023**, 013 (2024), arXiv:2310.03381 [hep-lat].

[44] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, HMC with Normalizing Flows, PoS **LATTICE2021**, 073 (2022), arXiv:2112.01586 [cs.LG].

[45] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation **14**, 1771 (2002).

[46] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, and R. Zemel, Learning the stein discrepancy for training and evaluating energy-based models without sampling (2020), arXiv:2002.05616 [stat.ML].

[47] D. C. Hackett, C.-C. Hsieh, S. Pontula, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan, Flow-based sampling for multimodal and extended-mode distributions in lattice field theory (2025), arXiv:2107.00734 [hep-lat].

[48] K. A. Nicoli, C. J. Anders, T. Hartung, K. Jansen, P. Kessel, and S. Nakajima, Detecting and mitigating mode-collapse for flow-based sampling of lattice field theories, Phys. Rev. D **108**, 114501 (2023), arXiv:2302.14082 [hep-lat].

[49] S. Chen, O. Savchuk, S. Zheng, B. Chen, H. Stoecker, L. Wang, and K. Zhou, Fourier-flow model generating Feynman paths, Phys. Rev. D **107**, 056001 (2023), arXiv:2211.03470 [hep-lat].

[50] J. Köhler, L. Klein, and F. Noé, Equivariant flows: Exact likelihood generative learning for symmetric densities

(2020), arXiv:2006.02425 [stat.ML].

[51] J. Ostmeyer and P. Buividovich, Minimal autocorrelation in hybrid monte carlo simulations using exact fourier acceleration, Comput. Phys. Commun. **313**, 109624 (2025), arXiv:2404.09723 [hep-lat].

[52] U. Wolff (ALPHA), Comput. Phys. Commun. **156**, 143 (2004), [Erratum: Comput.Phys.Commun. 176, 383 (2007)], arXiv:hep-lat/0306017.

[53] Q. Zhu, G. Aarts, W. Wang, K. Zhou, and L. Wang, Physics-conditioned diffusion models for lattice gauge theory (2025), arXiv:2502.05504 [hep-lat].

[54] R. Cavoretto and A. De Rossi, Adaptive refinement techniques for rbf-pu collocation, in *Large-Scale Scientific Computing*, edited by I. Lirkov and S. Margenov (Springer International Publishing, Cham, 2020) pp. 84–91.

[55] Y. Efendiev, J. Galvis, and T. Y. Hou, Generalized multiscale finite element methods (gmsfem), Journal of Computational Physics **251**, 116 (2013).

[56] E. T. Chung, Y. Efendiev, and W. T. Leung, Fast online generalized multiscale finite element method using constraint energy minimization, Journal of Computational Physics **355**, 450 (2018).

[57] R. Schaback and H. Wendland, Adaptive greedy techniques for approximate solution of large RBF systems, Numerical Algorithms **24**, 239 (2000).

[58] R. A. DeVore and V. N. Temlyakov, Some remarks on greedy algorithms, Advances in Computational Mathematics **5**, 173 (1996).

[59] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Computational Mathematics **4**, 389 (1995).

[60] E. J. Kansa, Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates, Computers & Mathematics with Applications **19**, 127 (1990).

[61] E. J. Kansa, Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations, Computers & Mathematics with Applications **19**, 147 (1990).

[62] U. Wolff, Critical slowing down, Nucl. Phys. B Proc. Suppl. **17**, 93 (1990).

[63] R. Abbott, M. S. Albergo, A. Botev, D. Boyda, K. Cranmer, D. C. Hackett, A. G. D. G. Matthews, S. Racanière, A. Razavi, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Aspects of scaling and scalability for flow-based sampling of lattice qcd (2022), arXiv:2211.07541 [hep-lat].

[64] L. Del Debbio, J. Marsh Rossney, and M. Wilson, Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability, Phys. Rev. D **104**, 094507 (2021).

[65] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Sampling using $SU(N)$ gauge equivariant flows, Phys. Rev. D **103**, 074504 (2021), arXiv:2008.05456 [hep-lat].

[66] M. Gerdes, P. de Haan, R. Bondesan, and M. C. N. Cheng, Continuous normalizing flows for lattice gauge theories (2024), arXiv:2410.13161 [hep-lat].

[67] G. Aarts, D. E. Habibi, L. Wang, and K. Zhou, Combining complex langevin dynamics with score-based and energy-based diffusion models (2025), arXiv:2510.01328 [hep-lat].

[68] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Equivariant flow-based sampling for lattice gauge theory, Phys. Rev. Lett. **125**, 121601 (2020), arXiv:2003.06413 [hep-lat].

[69] D. Bachtis, G. Aarts, F. Di Renzo, and B. Lucini, Inverse Renormalization Group in Quantum Field Theory, Phys. Rev. Lett. **128**, 081603 (2022), arXiv:2107.00466 [hep-lat].

[70] S. Efthymiou, M. J. S. Beach, and R. G. Melko, Super-resolving the ising model with convolutional neural networks, Phys. Rev. B **99**, 075113 (2019).

[71] A. Singha, E. Cellini, K. A. Nicoli, K. Jansen, S. Kühn, and S. Nakajima, Multilevel generative samplers for investigating critical phenomena (2025), arXiv:2503.08918 [cs.LG].

[72] K. Masuki and Y. Ashida, Generative diffusion model with inverse renormalization group flows (2025), arXiv:2501.09064 [cond-mat.stat-mech].